REMARKS

Claims 1-6, 8, 10, 12, and 14-30 have been cancelled. Claims 7, 9, 11, and 13 have been rewritten as independent claims.

Claims 7 and 9 stands rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,522,075 ("Robinson"). Claim 11 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Robinson in view of Rozycki. Claim 13 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Robinson in view of Rozycki in further view of Walker.

With respect to the rejection of claim 7 under 35 U.S.C. §102(b), the examiner argues that Robinson describes all of the limitations of claim 7, including, in lines 50-60 of column 12, reporting an ability of a processor to support said processor mode using one of a plurality of reserved feature bits that are returned in a processor register.

Lines 50-60 of column 12 of Robinson read:

> *As noted above, in the present invention sensitive instructions are those that can be executed only by the VMM. Thus, is a VM attempts to execute such an instruction, it traps to the VMM. This may be done through the use of the VM-bit. Whenever the processor attempts to execute a sensitive instruction, it first checks to determine if the VM-bit is set. If the VM-bit is set (i.e., if the processor is running in the VM mode) then the system traps to a process in the VMM where the VMM can emulate the requested sensitive instruction. Emulation routines for the VMM are generally known in the art and will not further be discussed.*

Contrary to the examiner's argument, the VM-bit of Robinson does not report an ability of a processor to support VM mode. Instead, the VM-bit of is checked to determine if the system should trap to a process in the VMM. In other words, the VM-bit not being set does not mean that the processor does not support VM mode, it means that the processor is not in VM mode.

Therefore, even if Robinson disclosed all of the other limitations of claim 7, and if the VM mode of Robinson was the same as the processor mode of claim 7, both of which applicants reserve the right to traverse, Robinson still would not anticipate claim 7.

Therefore, the applicants respectfully request the withdrawal of the rejection of claim 7.

With respect to the rejection of claim 9 under 35 U.S.C. §102(b), the examiner argues that Robinson describes all of the limitations of claim 9, including, in lines 35-40 of column 12, determining that the attempt of the guest software would succeed if the guest software was running outside said processor mode.

Lines 35-40 of column 12 of Robinson read:

> *There are various reasons that an instruction may be classed as sensitive. For example, as discussed above, any instruction that halts the execution of the processor is sensitive. Further, since in the present invention only the VMM can modify the address maps, any instruction that could be executed by a VM that could modify the address maps must also be classified as sensitive.*

Contrary to the examiner's argument, Robinson does not describe determining if an attempt of guest software would succeed if the guest software was running outside of Robinson's VM mode. Instead, Robinson gives examples of instructions that may be classified as sensitive. Robinson does not further describe a determination of whether these sensitive instructions would succeed if run in Robinson's VM mode; instead, Robinson describes trapping on all of these sensitive instructions.

In contrast, claim 9 recites determining if an attempt would succeed if the guest software were running outside the processor mode. Page 16 of the present application gives an example of such a determination – whether an RDMSR instruction is being attempted by guest software running with supervisor privilege. If so, the instruction would succeed if the guest software were running outside the processor mode. Robinson describes no such determination, by example or otherwise.

Therefore, even if Robinson disclosed all of the other limitations of claim 9, and if the VM mode of Robinson was the same as the processor mode of claim 9, both of which applicants reserve the right to traverse, Robinson still would not anticipate claim 9.

Therefore, the applicants respectfully request the withdrawal of the rejection of claim 9.

With respect to the rejection of claim 11 under 35 U.S.C. §103(a), the examiner argues that Rozycki describes identifying an attempt of the guest software to modify an interrupt flag, and modifying the interrupt flag if the interrupt flag does not control masking of interrupts.

The examiner argues:

> *Rozycki discloses a technique whereby a flag is checked and based on its value appropriate action is taken and where it is possible to execute some system level programs or procedures at the application level.*

However, what the examiner argues is disclosed by Rozycki is not the same as modifying an interrupt flag if the interrupt flag does not control masking of interrupts. What the examiner argues is disclosed by Rozycki relates to the value of a flag, not to what the flag controls. Rozycki does not describe taking any action based on whether an interrupt flag controls masking of interrupts.

Therefore, even if the combination of Robinson and Rozycki disclosed all of the other limitations of claim 11, and the combination were proper, both of which the applicants reserve the right to traverse, the combination still would not render claim 11 unpatentable.

Therefore, the applicants respectfully request the withdrawal of the rejection of claim 11.

With respect to the rejection of claim 13 under 35 U.S.C. §103(a), the examiner argues that Walker describes preventing the attempt of the guest software to modify the interrupt flag, and including providing a shadow interrupt flag for modifications by the guest software.

The examiner refers to page 7 of Walker, which describes shadow registers as follows:

> *Each time a shadowed register's value changes, the normal register's old value is saved in the shadow register. When an interrupt occurs, the normal register values can be restored from shadow registers to serialize the processor state. Shadow register values cannot be changed by an instruction until the processor is assured that the instruction will not cause an interrupt.*

The use of shadow registers as described by Walker is not the same as providing a shadow interrupt flag for modifications by the guest software. Walker uses shadow registers to save the old state of a normal register when the value of the normal register is changed. In contrast, claim 13 includes preventing guest software from modifying an interrupt flag and allowing guest software to modify a shadow interrupt flag. Therefore, even if the combination of Robinson, Rozycki, and Walker disclosed all of the other limitations of claim 13, and the combination were proper, both of which the applicants reserve the right to traverse, the combination still would not render claim 13 unpatentable.

Therefore, the applicants respectfully request the withdrawal of the rejection of claim 13.

CONCLUSION

Based on the foregoing, the applicants respectfully submit that the rejections of claims 7, 9, 11, and 13 have been overcome, and that claims 7, 9, 11, and 13 are in condition for allowance. The applicant therefore respectfully requests the issuance of a Notice of Allowance. Please charge any necessary fees to our Deposit Account No. 50-0221.

Respectfully submitted,

Date: June 22, 2005

Thomas R. Lane
Registration No. 42,781